



UNITED STATES PATENT AND TRADEMARK OFFICE

JO
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/053,045	01/15/2002	Ziv Z. Hellman	SVL920060523US1	3464
49330	7590	02/12/2007	EXAMINER	
DUKE W. YEE P.O. BOX 802333 YEE & ASSOCIATES, P.C. DALLAS, TX 75380			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2193	
SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE		
3 MONTHS	02/12/2007	PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)
	10/053,045	HELLMAN ET AL.
	Examiner Tuan A. Vu	Art Unit 2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 11/30/06.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-69, 135 and 136 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-69, 135 and 136 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>10/19/06; 1/9/07</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application
	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 11/30/2006.

As indicated in Applicant's response, claims 1, 34, 135 have been amended. Claims 1-69, 135-136 are pending in the office action.

Claim Objections

2. Claims 1, 34, and 135 are objected to because of the following informalities: the use of the term 'can transform' (cl. 1, lines 10, 12; cl. 34, line 13; cl. 135, li. 10-11) should be corrected to yield a patentable weight and/or to map with the teachings in the Specifications. According to which, the derived graphical information (Fig. 9-11; Specifications, pg. 23-49) resulting from mapping of data to a database or ontology model are used by the GUI/interface and the XSL support tool to generate XML/markup programmatic content or executable construct or SQL scripts (see Specifications, pg. 6, 7, 16 subitem (iv); pg. 22, li. 1-12). There is insufficient teaching from the above to establish the direct action of transforming or converting (from the derived data) as a direct (emphasis added) result of the mapping, according to which derived (*derived transformation*) tables are being depicted (e.g. Specifications, pg. 6, 7, 16 sub-item (iii)) at length in the Disclosure (tables of pg. 23-49). Such derived tables represent the claimed derived transformation(s). The use of tables (Specifications, pg. 23-49) to edit a script or a XSL file (Specs pg. 99-116) require another form of conversion which appears to be lacking or largely ineffectively conveyed in the above 'can transform' limitation. Further, the lack of weight conveyed by the 'can transform' limitation is not supportive of a true step action (which would fulfill a USC 101 Practical Application Test requirement), action that can be construed by via a derived SQL/XSL script which when executed would transform data as intended; therefore, does

Art Unit: 2193

not constitute an patentable certainty about a step limitation being taken. The claim as of now amounts to reciting that a *derived transformation* would be purported for transforming a first data into a second data, which seems obscure as to why a transformation viewed as mere tables can in turn become the entity potentially able to perform an active transformation otherwise expected from an executable file. First, it should be a converting step to enable the transformation to become an executable capable of transforming; and second the act of transforming has to be an explicitly carried out execution, not a static potentiality. The above terminology should be corrected lest its interpretation would lead onto the deficiency of the lack of enablement.

Claims 1, and 34, 135 are objected to because of the language reciting: 'deriving using the ontology model... a transformation for transforming' in conjunction with 'can transform ... *without* transforming the first data ...via an ontological format'. There is no clear definition of the phrase 'can transform ... without via an ontological format' in light of the ontology-oriented transformation/mapping process as disclosed in the Specifications (e.g. pg. 14-16). That is, according to which, the RDBS structures are extracted into basic elements displayed as chart or tables reflecting the database row/columns elements (DB format) and the transformation step (creation in terms of OO class or objects or markup constructs from which to apply further enhancement by the users) from source data into target data is based on passing through the stage of such mapping via extraction; whereby it is not reasonably conveyed that a format such as a ontological representations is not present during any stages (*via a ontological format*) leading to the transformation process, if any. From the claim, the transformation of the first data (into a target data) is construed as being implemented using an ontology model (Specifications, pg. 7, li.

Art Unit: 2193

20-24), and this is a stage whereby (via which) constructs of a database model are retrieved into tables. When there is no clear explicit definition in the disclosure as to what this ‘transform ... without transforming ... via an ontological format’ actually signifies, this *via ... ontological format* negative limitation would be unclear and syntactically improper. The Specifications mention that the ‘derived transformations ... go directly from source RDBS to target RDBS without having to transform data via an ontological model’ (Specs, pg. 6, lines 31-33; pg. 14, lines 10-12) and one skill in the art does not learn much from this broad phrase termed as ‘go directly ... from...’, especially when the Specs mention about transforming ‘using the ontology model’ (pg. 7, li. 20-24), and when there is no explicit statement reasonably enforcing that the ‘go directly ...from’ phrase signifies a actual transformation being executed by an SQL executable, in light of the tables derivation and editing of a script as set forth above. Besides, the ‘can transform’ limitation does not bear any weight as to whether a transforming action is really performed by any form of SQL script (or even by the derived tables); and this has been emphasized and set forth above. Absent a definition of a explicit step of transformation using -- or not using -- an ontology format, the disclosure lacks sufficient, deliberate and clear teaching as to how this non-use of ontological format has been implemented.

Claim 34 recites ‘wherein the derived transformation *transforms* the first data *directly* into the second data’ (lines 10-11); and according to the Specifications the derived tables (*derived transformation*) are to be used involving judicious intervention of the user in order to yield an additional script via another form of transformation. For the sake of argument, suppose that the transformation is derived as a script (which is not recited), the claim has to establish a step the make use of this script (e.g. executing the derived script) to carry out its functionality for

Art Unit: 2193

some data transformation can be tangibly perceived. Thus, the language of ‘derived transformation … transforms … directly’ would be improper syntactic usage because a output from a database mapping can not suddenly turn into a executing program actually yielding transformed data. Broad and reasonable interpretation has it that the claimed mapping (against a model) amounts solely to derived information such as tables, not necessarily as deriving an executable. And not until the script gets executed will data of the first schema be transformed into the data of the target schema; that is, all of these extra steps of actually converting based on the derived transformation and then executing this result thereof are omitted. This omission renders the use of the above language syntactically deficient, faulty in terms of enablement, and unsupportive of the Specifications, such that if not corrected, will become a lack of enabling description of the invention.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 1-7, 9-30, 32-33, and 135-136 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a “useful, concrete, and tangible result” be accomplished. An “abstract idea” when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the “useful arts” when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a “useful, concrete and tangible result”.

Specifically, claim 1 recites a method comprising mapping a source data schema and a target schema onto a ontology model, and deriving a derived transformation for transforming first data conforming to the source schema into second data conforming to the target schema; wherein this derived transformation can transform the first data into the second data without transforming via an ontological format.

As construed, the derived transformation remains an entity unclear in terms of format, or any physical construct being tangible or operable by hardware engine to act as a transforming entity, lacking any reasonable action (transforming) steps conveying that this derived transformation is actually performing any form of transformation at all. The ‘derived transformation’ recited along with ‘can transform’ does not establish sturdy evidence that a action of transforming by an engine or module is taking place. The claim for stopping at just having a derived transformation created for a intended use/function (*for transforming*) or potentially capable (*can transform*) of transforming a first data into a second data amounts to a unrealized potential function without tangible form thereof for externalizing this potentiality into a real world application action; and thus does not constitute a Practical Application because it does not convey a real-world transformation of data leading to a concrete, useful and tangible result. The claim for remaining in the realm of abstract concept is rejected for leading to a non-statutory subject matter.

Claims 2-7, 9-30, 32-33 for not remedying to the lack of actual usage of internal data to yield tangible result via Application action are also rejected for leading to an abstract non-practical concept.

Claim 135 recites receiving a source schema and a target schema, mapping each onto a ontology model, deriving a transformation for the intended use of transforming first data into second data conforming respectively to said source and target schema such that the derived transformation can transform the first data into the second data. The claimed steps stop at obtaining a derived transformation, which amounts to abstract set of derived information without tangible manifestation of any sort thereof in terms of application real-world usage and functional embodiment. As set forth in claim 1, the 'can transform' limitation bears no patentable weight because of its potentiality being statically present but not carried out into a actual execution. The claim as a whole lack reasonable teaching that an actual action is taken to yield a concrete, useful and tangible result, and remains a non-practical abstract idea and is rejected for leading to a non-statutory subject matter.

Claim 136 is also rejected for not curing to the lack of real-world action leading to a application level tangible result as required by the Practical Test Application.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 1, 34, and 135 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The reciting of 'can transform ... without transforming via an ontological format' is not conveyed clearly in the disclosure or the claim to convey to one skill in the art how transformation is done such that it necessarily obviate the stage using a ontological format. From

the Specifications, ontological structure is one model where relationship among elements are visible; and in view of the analysis from above in the claim Objections, the derived class objects depicted in the user graphical for the user to further modify the model does read on a format that is ontological, and based on the Specifications from Fig.1, this model enable transformation in the last step. It is therefore unclear as to how transformation of the source data into some form can be achieved when it is required that there is no ontological format involved in this transformation. One skill in the art would not be able to construe the claimed invention based on what appears to be incompatible teachings from claim interpretation and the very facts from the disclosure; and this has been mentioned in the above Claim Objection, so that as a whole the claims (1, 34, 135) would make it hard for one skill in the art to make use of the invention. The deficient limitation will be treated as though the derived information would support creation of script or the likes operable to effectuate data conversion.

Claims 2-68, and 136 are also rejected from not remedying to the lack of definiteness of the base claims.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

- (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1-8, 20, 24, 26-41, and 135-136 are rejected under 35 U.S.C. 102(e) as being anticipated by Wachtel, USPN: 6,847,947 (hereinafter Wachtel).

As per claim 1, Wachtel discloses a method for deriving transformations for transforming data from one data schema to another, comprising:

receiving a source data schema (e.g. Fig. 14) and a target data schema (e.g. Fig. 17 –

Note: receiving a XML message in order to process it again reads on receiving a target schema – see col. 15, lines 1-29), the source data being different from the target data (Note: samples of schemas in Fig. 14, 17 depicts distinct forms of XML);

mapping the source data schema into an ontology model; mapping the target data schema into the ontology model (e.g. Fig. 6-9; *xml, workflow, map* – col. 5, line 62 to col. 7, line 14 - Note: XML information/metadata when parsed by XPATH into a tree reads on a schema of definition); and

deriving a transformation (e.g. Fig. 8-9, col. 12, line 51 to col. 13, line 26 – Note:

Intelligent data assimilation system reads on automatic derivation) for transforming data conforming to the source data schema into data conforming to the target data schema (e.g. Fig. 4-5; Fig. 6-9; Fig. 17-18 – Note: derivation of class in a workflow – see Fig. 4-5 --view reads on derived transformation used for further transformation; and output response based on mapping against ontology metastore based on service and output requests – Fig. 14, 17 -- reads on source data schema in request conforming to target schema), using the ontology model;

wherein the transformation transforms data conforming to source schema directly to data conforming target schema (Note: using abstracted concepts stored in the model and invoking one or more pre-stored templates to instantiate a workflow – see col. 5, line 46 to col. 7, line 14 -

- to include intelligence enabling the transforming into a target schema reads on data schema

from the request --i.e. source schema -- directly converted into output schema without any

intermediate data schema format; Fig. 10, step 516, 518; Fig 6 and related text),

wherein the transformation can transform the first data into the second data without transforming the first data via ontological format (Note: providing mapping by Weblogic assimilation integrator – see col. 9 and Fig. 4-- to obtain atomic objects and derived objects grouping into a model (or ontological abstraction) or LSO-based workflow, enhancing this LSO collection with object oriented attributes – see Fig. 9 – so to generate XML form to fulfill a request – see col. 7, lines 4-14; col. 14, bottom -- **reads on** without transforming via any ontology format, i.e. using no intermediate format from the DBMS, from the atomic objects, no intermediate format from the extracted basic/atomic units generated by the provider Weblogic mapping process – see col. 9).

As per claim 2, see ontology (col. 5, line 62→ col. 7, line 14).

As per claim 3, see ontology (e.g. *runtime ...populated* - col. 7, lines 23-65; Fig. 4).

As per claim 4, see Wachtel for external format (e.g. *xml document*– col. 7, lines 4-22) into internal format (e.g. *atomic semantic instances* - Fig. 4).

As per claim 5, see generating model instances (*runtime ...populated* - col. 7, lines 23-65; Fig. 4)

As per claim 6, Wachtel discloses initial model (e.g. *data fields* – col. 9, lines 40-44; col. 13, lines 1-26) being populated into runtime (ontology workflow model) with abstraction instances based on LSO mapping (Fig. 6-9; col. 14, line 60 to col. 15, line 41)

As per claim 7, refer to claim 4.

As per claim 8, Wachtel discloses code for transforming data conforming to the source data schema into data conforming to the target data schema (e.g. Fig. 6-9; Fig. 17-18; col. 6, line 20 to col. 7, line 14 – Note: developers configuring workflows via dynamic manipulation with encapsulation of objects therein and underlying XSL code supporting the formatting of output response based on service – Fig. 14, 17 -- reads on generating executable code to support transformation source data schema in request conforming to target schema-- see: XSL - col. 25, lines 41-47).

As per claim 20 Wachtel discloses that the source data schema is a source document schema describing source documents (e.g. Fig. 14), and wherein the target data schema is a target document schema describing target documents (e.g. Fig. 17-18 – Note: XML schema reads on describing target documents being defined by such schema).

As per claim 24, Wachtel discloses wherein the source document schema is a source XML schema describing source XML documents, wherein the target document schema is a target XML schema describing target XML documents, and wherein the source XML schema and the target XML schema each describes at least one XML complexType having at least one XML element or XML attribute (e.g. Fig. 14, 18 – Note: type as defined in a corresponding DTD having at least one XML element or XML attribute reads on complexType).

As per claim 26, Wachtel discloses XSLT script (col. 7, lines 10-14).

As per claim 27, Wachtel discloses that said mapping a source data schema and said mapping a target data schema each comprise: identifying at least one class in the ontology model corresponding to at least one XML complexType; and identifying at least one property or

composition of properties in the ontology model corresponding to at least one XML element or XML attribute (*class* – col. 11, lines 45 to col. 12, line 19; Fig. 4-7).

As per claim 28, Wachtel discloses that automatically deriving further comprises expressing XML elements and XML attributes of the target XML schema in terms of XML elements and XML attributes of the source XML schema (e.g. Fig. 6-9; Fig. 17-18 – Note: output response based on mapping against ontology metastore based on service and output requests – Fig. 14, 17 -- reads on source data schema in request conforming to target schema, i.e. expressing elements required from source XML in terms of elements in the output XML).

As per claim 29 Wachtel discloses said expressing is performed recursively through XPath paths (Note: parsing a XML schema according to W3C standard requires a Xpath and a DOM tree traversal; hence the Xpath recursive traversal is disclosed).

As per claim 30, Wachtel discloses wherein at least one dependency exists among properties in the ontology model, and wherein said deriving further comprises translating the at least one dependency among properties in the ontology model as at least one dependency between target XML elements and source XML elements (e.g. Fig. 3, 4, 5, 7, 17).

As per claim 31, Wachtel discloses applying the XSLT script to at least one source XML document to generate at least one target XML document (re claim 26).

As per claims 32 and 33, Wachtel discloses single (Fig. 1) or multiple databases (Fig. 8).

As per claim 34, Wachtel discloses a system for deriving transformations for transforming data from one data schema to another, comprising:

a schema receiver receiving a source data schema and a target data schema (e.g. Fig. 14; Fig. 17 - Note: sub-workflow receiving a XML message in order to process it again reads on

receiving a target schema – see col. 15, lines 1-29), the source data being different from the target data (Note: samples of schemas in Fig. 14, 17 depicts distinct forms of XML being readable from a computer media otherwise the transformation would be of no use);

a mapping processor mapping a data schema into an ontology model (e.g. Fig. 6-9; *xml*, *workflow*, *map* – col. 5, line 62 to col. 7, line 14); and

a transformation processor deriving a derived transformation for transforming first data conforming to the source data schema into second data conforming to the target data schema, based on respective source and target mappings generated by said mapping processor (Fig. 6-9; Fig. 17-18 – Note: output response based on mapping against ontology metastore based on service and output requests – Fig. 14, 17 -- reads on source data schema in request conforming to target schema) for mapping said source data schema and said target data schema into a common ontology model;

wherein the transformation transforms data conforming to source schema directly to data conforming target schema (Note: see col. 5, line 46 to col. 7, line 14 -- to include external intelligence and supporting structures enabling the transforming into a target schema reads on data schema from the request --i.e. source schema -- directly converted into output schema without any intermediate data schema format; Fig. 10, step 516, 518; Fig 6 and related text), wherein source mapping and target mapping is based on a ontology model (Fig. 4 in light of the basic atomic DBMS objects);

wherein the transformation can transform the first data into the second data without transforming the first data via ontological format (Note: providing DBMS mapping by Weblogic assimilation integrator – see col. 9 and Fig. 4-- to obtain atomic objects, thereby deriving object

grouping into a model (or ontological abstraction) or LSO-based workflow, enhancing this LSO collection with object oriented attributes – see Fig. 9 – so to generate XML form to fulfill a request – see col. 7, lines 4-14; col. 14, bottom -- **reads on** without transforming via any ontology (or DBMS) format, i.e. using no intermediate format from the DBMS, from the atomic objects, no intermediate format from the extracted basic/atomic units generated by the provider Weblogic mapping process – see col. 9).

As per claims 35-41, these claims correspond to claims 2-8 respectively; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 135, Wachtel discloses an article of manufacture including one or more computer-readable media that embody a program of instructions for transforming data from one schema to another, wherein the program of instructions, when executed by a processing system, causes the processing system to:

receive (source data schema and a target data schema...);

map the source data schema (into an ontology model...);

map the target data schema (into the ontology model...); and

derive a derived transformation (for transforming data conforming to the source data schema into data conforming to the target relational database schema, using the ontology model); wherein the derived transformation ... can transform ... target data schema, wherein the derived ... without transforming ... via ontology format; all these limitations having been addressed in claim 1 above.

As per claims 136, see Wachtel Fig. 1, col. 30-31: *computer readable media - claims 39, 56, or 57.*

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 9-19, 21-23, 25, 42-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wachtel, USPN: 6,847,947, further in view of Lindberg et al., USPN: 6,732,109 (hereinafter Lindberg)

As per claims 9-10, Wachtel does not explicitly disclose that (re claim 9) wherein the source data schema is a source table schema describing source data tables, wherein the target data schema is a target table schema describing target data tables, and wherein the source table schema and the target table schema each describes at least one table having columns; and that (re claim 10) wherein the source table schema is a source relational database schema describing source relational database tables, wherein the target table schema is a target relational database schema describing target relational database tables. However, Wachtel discloses, from figures 3, 8, 13, that the automatic transformation is an SQL query in that Wachtel discloses a query system involving a database search (Fig. 3, 8, 13; DBMS – col. 9, lines 23-27) and a service to fulfill search in database based on XML specifications and then update database thereafter (*updates database tables* - col. 14, line 50 to col. 15, line 67) and database used in conjunction with modeling framework can be typified by Lindberg in which XML specifications can be mapped into properties relationships (col. 8, line 33 to col. 14, line 10). It would have been

obvious for one skill in the art at the time the invention was made to implement the XML schema as parsed by Wachtel in the assimilation workflow instance so that both the target and source schemas correspond respectively to the database tables as shown by Lindberg, having relational DB constructs because schema is a fundamental means by which a relational database is designed/implemented and XML schema used to update records of data as suggested by Wachtel would be of better use if such XML schema is such as it corresponds the database tables upon which Wachtel's method is to fulfill client queries -- to fetch DB data and for updating-- it as purported above by the RDBM of Lindberg and the update approach by Wachtel.

As per claim 11, Wachtel discloses creating class and properties (*class* – col. 11, lines 45 to col. 12, line 19) and identifying at least one class in the ontology model corresponding to at least one table; and identifying at least one property or composition of properties in the ontology model corresponding to at least one table column. The mapping of metadata/specification data with the database tables has been rendered obvious from claims 9-10 above.

As per claim 12, Wachtel based on the rationale of the rejection in claim 9-10; and the teaching from automatic deriving of classes and properties in claim 11, discloses deriving comprises: labeling properties of the ontology model with symbols; converting at least one column in the source relational database schema into at least one source symbol; converting at least one column in the target relational database schema into at least one target symbol; and expressing the at least one target symbol in terms of at least one source symbol.

As per claim 13, Wachtel discloses composition of properties (e.g. Fig. 4, 5, 7, 17).

As per claim 14, as for the automatic deriving, Wachtel does not explicitly discloses wherein at least one dependency exists among properties in the ontology model, and wherein

said deriving further comprises translating the at least one dependency among properties in the ontology model as at least one dependency between target relational database columns and source relational database columns, and wherein said expressing incorporates the at least one dependency between target relational database columns and source relational database columns. But in view well-known practices at the time the invention was made where relational databases are used with the construction in information model such as typified by Lindberg in which XML specifications can be mapped into properties relationships (col. 8, line 33 to col. 14, line 10), the concept of translating input schema to output schema based on such database-based dependency of data would have been inferred from the teachings by Wachtel as addressed in claims 9-10 using XML schema. Hence based on the teachings by Lindberg, it would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the XML schema mapping of Wachtel so that it is added with the relational data dependency as by Lindberg so that output schema derived from input schema is founded on the dependency of properties as set forth in the relational database tables because this would facilitate the developing of results and fulfill the requirements owing to the hierarchy of properties or layering of requirements thus enabling time efficient mapping and model developing as purported by Lindberg (see Background of invention) which lead to efficient gathering of required data to produced the desired output result for the DB query by Wachtel.

As per claims 15-16, Wachtel does not expressly disclose wherein said expressing uses expressions involving arithmetic operations and that said expressing uses expressions involving character string operations. But the query operations to implement a search as by Lindberg or

intended by Wachtel would have inherently encompassed operations using character string or arithmetic/logical operator to query a specific data record.

As per claim 17, Wachtel discloses applying the query to at least one source relational database table to populate at least one target relational database table (e.g. Fig. 3, 8, 13).

As per claims 18-19, Wachtel discloses single (Fig. 1) or multiple databases (Fig. 8).

As per claim 21, Wachtel does not explicitly discloses that the source document schema is a source DTD describing source XML documents, wherein the target document schema is a target DTD describing target XML documents, and wherein the source DTD and the target DTD each describes at least one XML element or XML attribute. Official notice is taken that at the time the invention was made a XML document and its attributes being accompanied with definition file such as a DTD document was a well-known concept. Hence in case Wachtel's XML schema definition does not already include such DTD schema, it would have been obvious for one skill in the art at the time the invention was made to provide such DTD schema along with the XML documents so that both source XML and target XML are defined according to the well known concept because without definition the attributes and properties of the XML would not be proper to be parsed by the XML parsing technologies as implemented by W3c standards.

As per claims 22 and 23, Wachtel discloses the transformation is using J2EE query/messaging service (e.g. col. 9, lines 8-26) with support of XML but does not expressly mention Xquery but this limitation would have been obvious because based on the J2EE and BEA framework, query against RDMS using Java would have been more beneficial if implemented with Xquery using the technology based on XML structures and its metastore as disclosed by Watchtel (Fig. 1) and Wachtel discloses XSLT script (col. 7, lines 10-14)

As per claim 25, the Xquery limitation would have been obvious by virtue of the rationale set forth in claim 22.

As per claims 42-44, these claims correspond to claims 9-11 respectively; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 45, Wachtel discloses presents a user interface to enable workflow assembling and application of integration rules using repository data (Fig. 1; *DBMS* – col. 9, lines 23-27) but does not explicitly disclose wherein said property identifier presents a user with a choice of at least one property in the common ontology model that may correspond to a given table column even though a database entails representing persisted objects by table. The presentation of properties using a model to map such property for a query against a column in a database has been addressed above using Lindberg (e.g. col. 7, line 47 to col 14, line 17). In view of the rationale as set forth in claim 9-10 and the user interaction as presented in a ontology model as endeavored by Wachtel, this limitation would have been obvious for the same reasons as set forth in claims 9-10, because of the purpose of ontology-based modeling is to allow user to customize requirements of a application based on some schema mappings and these are both disclosed in Wachtel and Lindberg.

As per claims 46 and 47, in view of rationale in claim 42 and the class and properties being identified as to correspond to given column or a target table (Note: a key for a table is inherent in DB construct for facilitating query and normalization of data), as addressed in claim 11, these claims are rejected with the rationale as set forth therein.

As per claim 48, Wachtel discloses labeling properties with symbols (see Fig.6; object 112 – Fig. 3); and combined with the teaching by Lindberg as set forth in the rationale

addressing claims 42-43, has rendered obvious converting at least one column in the source relational database schema into at least one source symbol, and converting at least one column in the target relational database schema into at least one target symbol; and expressing the at least one target symbol in terms of at least one source symbol (Note: the transformation as addressed in claim 34 combined with Lindberg would make it obvious the converting column/symbol limitation).

As per claims 49-52, and 54-55, these claims correspond to claims 13-16, 18-19 respectively; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 53, Lindberg discloses mapping applications to specific constructs of relational DB table (col. 7, line 47 to col 14, line 17) and Wachtel discloses mapping with eventual update of database (col. 14, line 50 to col. 15, line 67). It would have been obvious for one skill in the art to implement the mapping and transformation of Wachtel using database update such that applying the query processing by Lindberg so to retrieve one source relational database table; and applying the query to the at least one source relational database table to populate at least one target relational database table according to the suggested update technique by Wachtel because without update data on record would not be appropriate for subsequent queries and might generate data errors.

As per claims 56-69, these claims correspond to claims 20-33 respectively; hence are rejected with the corresponding rejections as set forth therein respectively.

Response to Arguments

11. Applicant's arguments filed 11/30/06 have been fully considered but they are not persuasive. Following are the Examiner's observations in regard thereto.

Claim Objections:

(A) The amendment now replacing 'adapted to' with 'can' is deemed insufficient because it does not establish a clear execution being taken by a entity reasonably able to actually transform data; and this objection is maintained and set forth above.

(B) Applicants has submitted that (Appl. Rmrks, pg. 13-14)'derived transformation' exist throughout the Specifications; that transformation is like a mathematical function; that the deriving the transformation is using the ontological model but that the transformation itself is not. The term 'derived transformation' has been recited in conjunction with mapping against a ontology model; and deriving data against a model can only yield additional data. A derived set of information cannot on its own constitute any executable engine like a Math function in the midst of executing to yield data. For the sake of argument, even if the data transformed by a Math function is an end-result, that mere form of data on its own remains in the abstract realm of processor/computer internal volatile environment activity, not a patentable weight. But the derived data from the recited mapping are interpreted in light of the Specifications (depicted in length) as mere descriptive non-functional entities, e.g. tables. Whether the derived transformation can transform or actually transforms any data does not fix the basic problem: a transformation from a mapping (emphasis added) using the model and yielding informative data cannot on its own become an executable (suddenly executing) without reasonable teaching to that effect; and the omission of such otherwise enabling details has been mentioned in the Claim

Objections. The argument about extensive disclosure in regard to the ‘derived transformation’ is thus not convincing.

(C) Applicants have submitted that (Appl. Rmrks, pg. 15) derived transformation is represented mathematical terms; and it can be shown in the Specifications pg. 1-5 as such and in pg. 14, li. 10-11. As set forth in the Claim Objection, there is not sufficient teaching concerning this ‘go ...directly from ... without having to’ phrase any where in the disclosure to corroborate that the table information derived from the ontology database can actually transform data directly from one RDBS format onto another RDBS format, and so doing, bypassing any transformation whatsoever using a format represented by the ontology model constructs. The argument is therefore not convincing for mixing up BACKGROUND teaching (Specs, pg. 5) with a very broad statement (Specs, pg. 14) without considering the very detailed implementation in the disclosure concerning how the script is generated in order to provide the targeted (e.g. RDBS) format transformation (which appears to be the crux of the claim)

35 USC §112 rejection:

(D) Applicants have submitted that ‘the transformation itself does not use an ontological format when transforming data’ (Appl. Rmrks, pg. 17, top para). This transformation has been interpreted as a informative data tabulated from mapping against a database; and this in reasonable terms cannot be equated to a Mathematical function without specifics in the claim conveying the executable nature of such transformation. As transformed, a set of data has to be reasonably conveyed as having a format, whether metastatic, tabular, or tangible substance, persistable or executable or programmatic entities. A *derived transformation* by no means cannot be given a entity (that turns into a executable form) that performs a operation just by

Art Unit: 2193

virtue of merely reciting such terms as ‘transforms’ or ‘can transform’ without substantial support by further claimed details, even taken in light of the Specifications, which in this case yields extensive description of tables; i.e. *tables* standing for what this derived transformation amounts to. The argument is then not convincing in light of the earlier observations to the effect that one skill in the art would find it hard to interpret the derived tables as being an binary or a executable script – as set forth also the Claim Objections.

35 USC §102 Anticipation:

(E) Applicants submitted that Wachtel teaches using ontological model definition at various level of abstraction to reach semantic object or atomic units; and using these atomic objects to create classes; and this is opposite to the claimed ‘derived transformation’ that transforms data without using an ontological format (Appl. Rmrks, pg. 21, top para). As set forth in the Claim Objection, the use of ‘derived transformation’ that transforms is improper language; a language which not only is syntactically deficient but does not reflect the semantic aspect of the Disclosure in its details. The above limitation has been treated as though some in-between conversion enables the derived information (from the mapping) to effect the data transform (Note: transformation as between two format or two database schemas is explained why Wachtel’s system is called Intelligent Assimilation system between multiple and heterogenous systems of diverse format); such that Wachtel’s XML response being sent to the client would stand for this executable form enabling this data conversion/transforming. It is also noted that the present Invention also describes gathering parameters, variables inside derived tables, and based thereupon, insert appropriate variable constructs inside some (markup language) programmatic entries/formulation leading to a executable script (SQL queries). Both the SQL

queries as gathered from the Invention (see Specifications: pg. 50, pg. 77-93) or by Wachtel's XML script enable execution to provide queries to data of different format, when the very use of the mapping against a database model is for the sake of creating portable and executable form in the likes of XML or markup language, such as endeavored by both Wachtel and the current Invention. If executing the script to effectuate data transformation or query of DBs objects require no more direct referral to actual ontology variables, then what the Invention is thought to achieve does not distinguish from Wachtel's use of XML response for the client to effect such data queries or transformation, because when executed, any executable script will perform data transformation. The 'derived transformation', if it were given the meaning of a script, is therefore at best performing nothing opposite to what is recited in Wachtel, such being a XML interpretable/executable script. The execution thereof, even though not recited, necessarily entails the connotation that it will be performed (say by the client machine as in Wachtel) without having to directly refer to a remote (ontological) database record of persisted data, variables; simply because the mapping by Wachtel's assimilation system has earlier extracted the derived level of abstracted units, objects/templatized data or workflow needed to create said XML response(s). In view of the Claim Objections and USC 112 rejections, the 'without transforming ... via an ontological format' is not given a proper patentable weight in view of lack of proper corroboration from the Disclosure. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. The argument is therefore non persuasive and Wachtel'

Art Unit: 2193

s creation of XML response by ways of the workflow garnered from the atomic objects obtained from a mapping has fulfilled the above “without ... via an ontological format’ limitation.

(F) Applicants maintained that Wachtel creating of XML formats using XSL sheets does use ontological definition (Appl. Rmrks, pg. 22, 2nd para) and that the Examiner’s quotations are taken out of context because using ontological definition is not what is claimed. Again, the claimed ‘derived transformation’ is not equated to an executable program being actually executed to yield data transformation, as set forth above in the Claim Objections. Because of the deficiency in the claim in terms of allowing the ‘derived’ data to execute a transformation, the limitation has been treated as using a script to do the data transform. If such ‘derived transformation’ were a script, it falls into the scenario of the markup script by Wachtel being sent to the client for the client to invoke the query calls based on such script. Invoking the calls inside an interpreted script will not necessitate a direct use of ontological definition; since these DB definitions after they were extracted from Wachtel’s mapping using the Assimilation module have been converted to script constructs; and these constructs have sufficed for the script to be used on its own, when sent back as a response. Besides, such script is created almost the same manner in the current Application’s Disclosure. The Applicants ‘ arguments are not convincing, first because the ‘derived transformation’ are mere tables; second, if they were executables, they have to be conveyed as being executed; and even so, they do not distinguish from Wachtel’s XML responses.

(G) Applicants submitted (Appl. Rmrks, pg. 23) that the LSOs by Wachtel are encapsulated objects and founded based on ontological semantics or format, which is not ‘transform ...

without ... via an ontological format'. This argument falls under the same ambit of the issues raised in section E or F above; hence is therefore non-convincing for the same reasons.

(H) The argument (Appl. Rmrks, pg. 25) that Wachtel's workflow uses ontological definition does not anticipate the above 'transform ... without ... via an ontological format' will be referred to section F, using the Claim Objections as set forth earlier and applied throughout the Office Action. The ontological model contains definition; and while the mapping is used to map data for both schemas into a form that can be used to put together a script, the use of such ontological model is inherent to the 'derived transformation' or tables (as in the current Invention) or creating of workflows (as in Wachtel). The script when created is operable via its execution to yield transformed data of one format into another; such that the execution relies solely on the constructs inside the script wherein these programmatic constructs bear their syntactic or semantic implication directly from the workflow (as in Wachtel's Fig. 4) or the tables data (as in the Invention: pg. 23-116); that is, the constructs of the script bear their being indirectly from the extraction based the ontological DB whereas the execution of the script (data transformation) is a executing entity dependent solely of the runtime environment. As executed, the data being transformed by the script action (e.g. at the client receiving Wachtel's XML response) is not directly associating any DB definition as persisted in the ontology model; which seems far at the extracting end of a server. Wachtel teaches the same approach as the current Invention in terms of extracting data and use it to assemble a markup script, the execution of such script would then constitute some data transformation. Broadly interpreted given the deficiency of the claimed language, Wachtel has fulfilled the limitation of this transformation provided by the 'derived transformation'.

35 USC §103 Rejection:

(I) Applicants have submitted that there is no motivation to combine (Appl. Rmrks, pg. 27).

In response, Wachtel's use of ontological model to enable scalability falls in the same endeavor as Lindberg's as to provide abstraction via a model for implementing complex business logic, hence alleviating recreating of code anew for complex systems just by reusing persisted levels of abstraction in a ontology DB. One skill in the art would see that addressing Web scalability and reusability via Wachtel's ontological model would be analogous to making abstracted database of knowledge for implementing complex and growing business transactions. The argument about non-analogous endeavor between Wachtel and Lindberg is largely insufficient to overcome the rationale to combine; and amounts to mere allegation without pointing out exactly how the combined teachings when effectuated in the rejection would yield undeniable negative teachings.

The claims will stand rejected as set forth in the Office Action.

Conclusion

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

Art Unit: 2193

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,

Application/Control Number: 10/053,045

Page 29

Art Unit: 2193

Art Unit 2193
February 7, 2007